Listing of Claims

1   Claim 1 (Currently Amended): A method of implementing atomic transactions in a

2 system, said method comprising:

3   requesting in a <u>user</u> program ~~logic~~ a transaction identifier for an atomic transaction~~,~~

4 ~~wherein said program logic is contained in a user program designed by a programmer~~;

5   generating said transaction identifier in a transaction manager in response to said

6 requesting;

7   specifying in said ~~program logic~~ <u>user program</u> a plurality of combinations ~~for execution~~

8 ~~in a sequential order~~, wherein each of said plurality of combinations contains said transaction

9 identifier, a task procedure, and a rollback procedure, wherein said task procedure implements

10 a part of said atomic transaction and said rollback procedure is designed to rollback said task

11 procedure;

12   executing <u>a set of</u> ~~said~~ task procedures in ~~said~~ <u>a</u> sequential order <u>according to said user</u>

13 <u>program, wherein said set of task procedures are contained in said task procedures specified</u>

14 <u>in said plurality of combinations</u>;

15   keeping track of ~~said~~ <u>a set of</u> rollback procedures ~~in said transaction manager~~

16 <u>corresponding to said set of task procedures, each of said set of procedures being determined</u>

17 <u>based on a combination corresponding to an executed task procedure contained in said set of</u>

18 <u>task procedures, said combination being contained in said plurality of combinations specified</u>

19 <u>in said user program</u>; and

20   executing said <u>set of</u> rollback procedures in a reverse order of said sequential order if

21 said atomic transaction is to be aborted,

22   <u>wherein said rollback procedure is specified as a separate procedure from said task</u>

23 <u>procedure in said user program</u> ~~wherein said rollback procedures are identified according to~~

24 ~~said keeping~~.

1   Claim 2 (Original): The method of claim 1, wherein said transaction identifier is

2 unique to each of the atomic transactions.

1   Claim 3 (Previously Presented): The method of claim 1, wherein said keeping

2 comprises storing data representing said rollback procedures in a stack.

1   Claim 4 (Original): The method of claim 3, wherein said stack is stored in a memory.

1   Claim 5 (Original): The method of claim 1, further comprising examining a status

2   returned by execution of one of said task procedures and performing said aborting if said

3   status indicates an error.

1   Claim 6 (Original): The method of claim 1, wherein said aborting is performed

2   asynchronously.

1   Claims 7 (Currently Amended): A computer readable medium carrying one or more

2   sequences of instructions representing a user program ~~logic~~ for execution on a system, said

3   user program ~~logic~~ implementing an atomic transaction, wherein execution of said one or

4   more sequences of instructions by one or more processors contained in said system causes said

5   one or more processors to perform the actions of:

6   requesting an identifier in said user program from a transaction manager for said

7   atomic transaction, wherein said transaction manager generates a unique value as said

8   identifier;

9   setting a variable to equal said identifier;

10   specifying a plurality of combinations in said user program for execution in said

11   system, wherein each of said plurality of combinations contains said variable ~~transaction~~

12   ~~identifier~~, a task procedure, and a rollback procedure, wherein said task procedure implements

13   a part of said atomic transaction and said rollback procedure is designed to rollback said task

14   procedure, wherein said variable in each of said plurality of combinations specifies said

15   identifier generated by said transaction manager; and

16   aborting said atomic transaction by specifying said identifier associated with an abort

17   procedure to cause said rollback procedures to be executed,

18   wherein ~~said program logic, including~~ said plurality of combinations~~,~~ and said abort

19   procedure are contained in ~~a~~ said user program ~~designed by a programmer~~.

1   Claim 8 (Original): The computer readable medium of claim 7, wherein said specifying

2   comprises including each of said plurality of combinations in a single procedure call.

1   Claim 9 (Original): The computer readable medium of claim 7, further comprising

2   examining a status returned by execution of one of said task procedures and performing said

3   aborting if said status indicates an error.

1   Claim 10 (Currently Amended): A computer readable medium carrying one or more

2   sequences of instructions for supporting implementation of an atomic transaction in a system,

3   wherein execution of said one or more sequences of instructions by one or more processors

4   contained in said system causes said one or more processors to perform the actions of:

5   generating an identifier for said atomic transaction for a user program;

6   receiving a plurality of combinations for execution from said user program, wherein

7   each of said plurality of combinations contains said transaction identifier, a task procedure,

8   and a rollback procedure, wherein said task procedure implements a part of said atomic

9   transaction and said rollback procedure is designed to rollback said task procedure;

10   executing a set of task procedures in a sequential order according to said user program,

11   wherein said set of task procedures are contained in said plurality of combinations ~~said task~~

12   ~~procedures~~;

13   keeping track of a set of rollback procedures corresponding to said set of task

14   procedures, each of said set of procedures being determined based on a combination

15   corresponding to an executed task procedure contained in said set of task procedures, said

16   combination being contained in said plurality of combinations specified in said user program;

17   and

18   executing said set of rollback procedures in a reverse order of said sequential order in

19   response to receiving an abort request,

20   wherein said rollback procedure is specified as a separate procedure from said task

21   procedure in said user program.

22   ~~wherein each of said plurality of combinations is received from a corresponding user~~

23   ~~program and wherein the roll back procedure received from a first user program is different~~

24   ~~from the roll back procedure received from a second user program.~~

1   Claims 11 - 12 (Canceled)

1   Claim 13 (Currently Amended): The computer readable medium of claim 10 ~~12~~,
2   wherein said transaction identifier is generated to be unique for each atomic transaction.

1   Claim 14 (Currently Amended): The computer readable medium of claim 10 ~~12~~,
2   wherein said ~~data is~~ set of rollback procedures are represented in the form of a stack.

3   Claim 15 (Original): The computer readable medium of claim 14, wherein said stack
4   is stored in a memory.

1   Claim 16 (Currently Amended): A computer system comprising:
2   a memory storing a plurality of instructions; and
3   a processing unit coupled to said memory and executing said plurality of instructions
4   to support implementation of atomic transactions in a programming environment, said
5   processing unit being operable to:
6   request in a user program ~~logic~~ a transaction identifier for an atomic
7   transaction;
8   generate said transaction identifier in a transaction manager in response to said
9   requesting, wherein said transaction manager is provided external to said user
10   program;
11   specify in said user program ~~logic~~ a plurality of combinations ~~for execution in~~
12   ~~a sequential order~~, wherein each of said plurality of combinations contains said
13   transaction identifier, a task procedure, and a rollback procedure, wherein said task
14   procedure implements a part of said atomic transaction and said rollback procedure is
15   designed to rollback said task procedure, wherein said rollback procedure is specified
16   as a separate procedure from said task procedure;
17   execute ~~said~~ a set of task procedures in ~~said~~ a sequential order according to said
18   user program;
19   keep track of ~~said~~ a set of rollback procedures ~~in said transaction manager~~
20   corresponding to said set of task procedures, each of said set of procedures being

21   determined based on a combination corresponding to an executed task procedure

22   contained in said set of task procedures, said combination being contained in said

23   plurality of combinations specified in said user program; and

24   execute said set of rollback procedures in a reverse order of said sequential

25   order if said atomic transaction is to be aborted, wherein said rollback procedures are

26   identified according to said keeping,

27   ~~wherein said program logic is contained in a user program designed by a~~

28   ~~programmer.~~

1   Claim 17 (Original): The computer system of claim 16, wherein said transaction

2   identifier is unique to each of the atomic transactions.

1   Claim 18 (Previously Presented): The computer system of claim 16, wherein said

2   processing unit is operable to store data representing said rollback procedures in a stack to

3   perform said keep.

1   Claim 19 (Original): The computer system of claim 18, wherein said stack is stored in

2   a memory.

1   Claim 20 (Original): The computer system of claim 16, wherein said processing unit

2   is further operable to examine a status returned by execution of one of said task procedures

3   and to perform said aborting if said status indicates an error.

1   Claim 21 (Previously Presented): The computer system of claim 16, wherein said

2   processing unit is operable to execute said rollback procedures asynchronously.

1   Claims 22 - 24 (Canceled)

1   Claim 25 (New): The computer readable medium of claim 7, wherein said rollback

2   procedure is specified as a separate procedure from said task procedure in said user program.